Design of Fractal Image Compression on SOC

A. JEDIDI, B. REJEB and M. ABID
CES, National school of engineer, University of Sfax, Tunisia.
E-mail: ahmedjedidi@yahoo.fr

Abstract

The technological revolution during the last years has carried out a great evolution especially in the multimedia domain. Nowadays, almost everyone benefits from various video applications such as TV broadcasting and video conferencing. This progress involves particularly an increase in the capacity of digital data transmission. As a result, data compression became increasingly significant for storage and transmission.

In this paper, we present an algorithm for fractal image compression based on SOC in real time. The algorithm consists of a hardware and software part. The hardware part supports an expensive calculation; therefore it is conceived on RTL level. The coding algorithm was implemented on the Altera chart STRATIX-I. The functional blocks were implemented with clock rate of 410 MHz with a maximum flow data input of 13.2 Gbit/s.

Index Terms— **Fractal image compression, SOC, FPGA, Real time, RTL level design, Functional simulation, Temporal simulation.**

1. Introduction

THE fast development of the data-processing applications was accompanied by a significant increase in the use of the digital images, in particular in the field of the multimedia, the plays, the transmission satellites or the medical imagery.

Due to their large size, digital images raise many problems for their transmission and storage. For example an image 512x512 pixels (value of the pixel is 8 bits) is necessitates 2MB of memory capacity. Also the real time which becomes more and more requires nowadays then the compression of image is necessary.

Currently several methods of image compression are proposed (JPEG, wavelets, etc.) in order to have a good compromise between the compression ratio and the quality of rebuilt image. Worthly mentioning is the **fractal image coding** method which is based on the fractal theory. The basic idea of this method is that the image can be reconstructed using its self similarity features.

it is noted that the phase of integration of the functionalities offered by the multimedia applications in the systems embarked in real time (i.e. the systems which carry out several calculations within a very limited time such as the portable telephones, TV, reader DVD etc) became increasingly complex [1], [2] and [7]. The implementations of the compression methods require the performance in term of computing power, flexibility, cost and time to market.

In this paper, we propose a hardware software design of fractal image compression. First, we briefly present the concept of hardware software fractal coding. Then, we mention internal architecture of hardware blocks. After that, we present the results of simulation and implementation of the system in Startix chard. Finally, we conclude the paper and we present open direction for future work.

2. Fractal coding

To understand quantitatively the principal encoding algorithm, consider the flow chart of the fractal image coder as shown in Figure 1. Encoding algorithm is composed of two main parts: software and hardware. The hardware part is called self similarity search [3] and [5].

The software part is made up four blocks: image partition, domain blocks classification, domain blocks reduction and range blocks classification [8] and [9]. This part is realizable in software and offers the real time criterion [9] and [10].

The self-similarity (hardware block) search between range and domain blocks is ensured by the Mean Square Error:

$$MSE = \frac{1}{B^2}\left(\sum_{1\le i,j\le B}\left(R_{i,j} - (\alpha D_{i,j} + \beta)^2\right)\right) \qquad \textbf{(1)}$$

Where:

- ♦ $R_{i,j}$ and $D_{i,j}$ are the pixels values of the range and domain blocks at coordinates (i,j).
- ♦ α and β are the contrast and the offset parameters.

978-0-7695-3170-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ISVLSI.2008.26

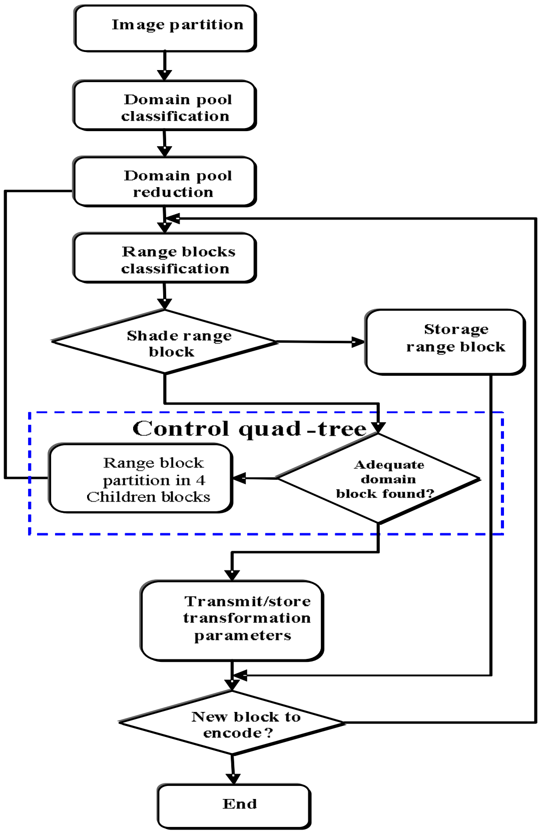
Image partition
Domain pool classification
Domain pool reduction
Range blocks classification
Shade range block
Storage range block
Control quad -tree
Range block partition in 4 Children blocks
Adequate domain block found?
Transmit/store transformation parameters
New block to encode?
End

Figure 1: Encoding algorithm flowchart

♦ B is the blocks width and height.

The optimal set of parameters for a given couple of range and domain blocks is found setting the partial derivates according to the contrast parameter α and the brightness β to zero [3].

$$\frac{\partial MSE}{\partial \alpha} = 0 \text{ and } \frac{\partial MSE}{\partial \beta} = 0 \qquad \textbf{(2)}$$

Finally the optimal offset and optimal contrast parameters are given by:

$$\alpha_{opt} = \frac{C_{DR}(0,0)}{\sigma_D^2} \qquad \textbf{(3)}$$

$$\beta_{opt} = \mu_R - \alpha_{opt} . \mu_D \qquad \textbf{(4)}$$

Where:

- μ_R and μ_D are respectively the mean values of the range R and domain D,

$$\mu_R = \frac{1}{B}\sum_{i=1}^{B} r_i \ ; \ \mu_D = \frac{1}{B}\sum_{i=1}^{B} d_i \qquad \textbf{(5)}$$

- C_{DR} is the covariance of R and D,

$$C_{DR} = \frac{1}{B}\sum_{i=1}^{B}(r_i - \mu_R)(d_i - \mu_D) \qquad \textbf{(6)}$$

- σ_D is the variance of D

$$\sigma_D = \frac{1}{B}\sum_{i=1}^{B}(d_i - \mu_D)^2 \qquad \textbf{(7)}$$

The computational complexity of the optimal parameters calculation is of order of o(N), where N is the block size.

The computational requirements, i.e. the number of additions, multiplications and divisions needed for the implementation of the optimal contrast parameter α_{opt} for blocks size N using the equation (3) are amounted to: 5N additions, 3N+1 multiplications, 2 subtractions and 7 divisions.

According to the equation (4), the number of operations needed for the implementation of the optimal offset parameter β_{opt} using the pre-calculated α_{opt} is amounted to one multiplication and one subtraction.

In the same way the Mean Square Error for a defined couple of range and domain blocks can be calculated using equation (1). The computational requirements are: N additions, 2N multiplications, N subtractions and 1 division

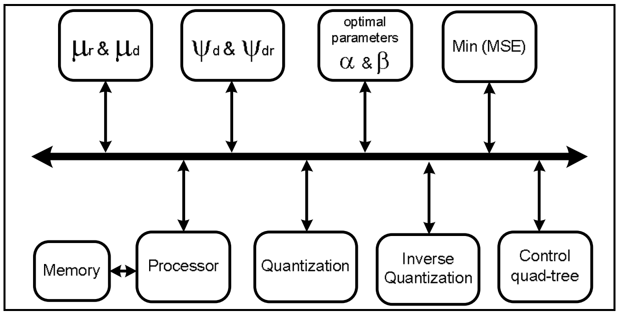

Figure 2: Architecture of SOC

Figure 2 presents the basic architecture of our chip. It is composed by: processor, memory and hardware dedicate blocks. The units communicate between them by means of a bus [6] and [8].

The hardware blocks are composed by seven elements:

1- Block μr &μd: is calculate the mean values domain and range.
2- Block Ψd &Ψdr: is to calculate the mean of product respectively domain/domain and domain/range
3- Optimal parameters α & β: its objective is to calculate the optimal parameters
4- Quantization and Inverse quantization: the first block is quantized the optimal parameters while the second block is their un-quantized.
5- Block min(MSE): this block calculate the minimum of mean square error

6- Control quad-tree: this block is responsible to control the system of quad-tree.

3. Design of the self-similarity search:

As already mentioned, self-similarity search is implemented in hardware dedicates blocks. One of the serious problems is the synchronisation of the data flow between the different hardware blocks. In order to satisfy this requirement, we are created a global State machine. Let's start by developing the different hardware dedicates blocks.

A. Hardware dedicates blocks architecture:

The proposed architecture for self-similarity search is shown in figure 3. It is composed by four stages implemented by serial-parallel architecture: average value, variance & covariance, the optimal parameters and validation stage.

i. Stage of average value: it is composed by four blocks. Each block is responsible to calculate: mean value of domain, mean value of range, mean product of domain and mean product of domain & range. These values are respectively denoted by: μ_r, μ_d, ψ_d and ψ_{dr} .
ii. Stage of variance & covariance: this stage contains two modules. First, the variance it's role calculates the square of variance of domain. That the Second, the covariance is calculating the covariance of range and domain. These different blocks are controlled by start_COV and start_VAR.
iii. Stage of optimal parameters: it is composed by two modules optimal contrast parameter and optimal offset parameter.
iv. Stage of validation: this stage is responsible to validate the different optimal parameters by using minimum (MSE).

B. Global State Machine:

The global architecture is controlled by different states machines. These states machines have a very significant role in this module self-similarity search.

For simplicity of the design of the state machine, it is important to divide it into two modules: Global State Machine and six sub states machines: fsm_M, fsm_COV, fsm_VAR fsm_Contarst, fsm_Offset and fsm_MSE, comes respectively in these blocks: average value, variance, covariance, optimal contrast parameter, optimal offset parameter and validation stage.

A block diagram of a global state machine is shown in figure 5. It is responsible to generate the different starts signals of six children states.

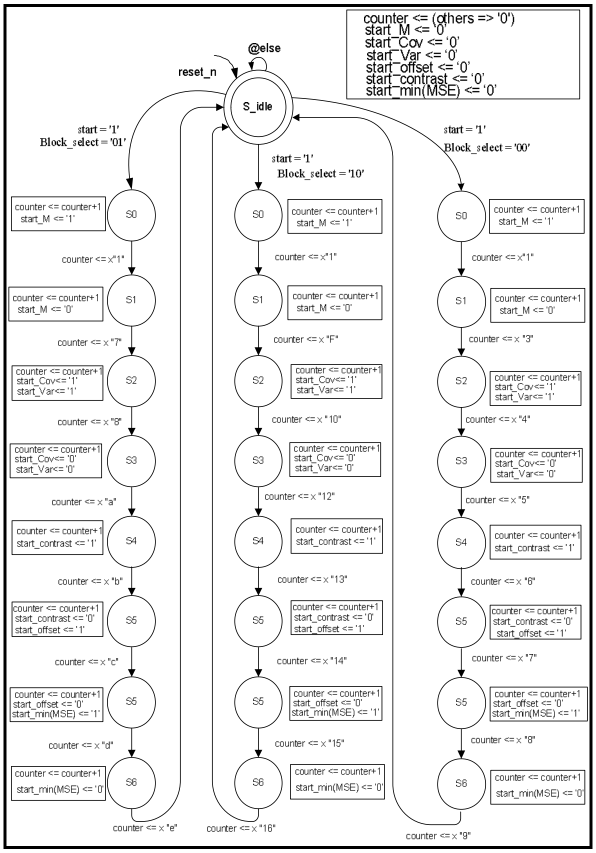

Figure 5: Global State Machine

4. Implementation results

To implement the two large modules: software and hardware (self-similarity search) we used chart FPGA STRATIX of the Altera firm which uses processor NIOS.

For architecture self-similarity search we propose several designs on RTL level in order to have a synthesisable architecture which offers a good compromise between the frequency, the area, the time execution and the data flow. The presented architecture has been simulated using Active VHDL. Thus, we make a functional simulation to validate the correct operation of this architecture. Then we pass to the temporal simulation which makes it possible to guarantee an implementation success of 95% on the STRATIX FPGA chard [4].

The total implementation in FPGA for the fractal image coding on the STRATIX chard offers these results:

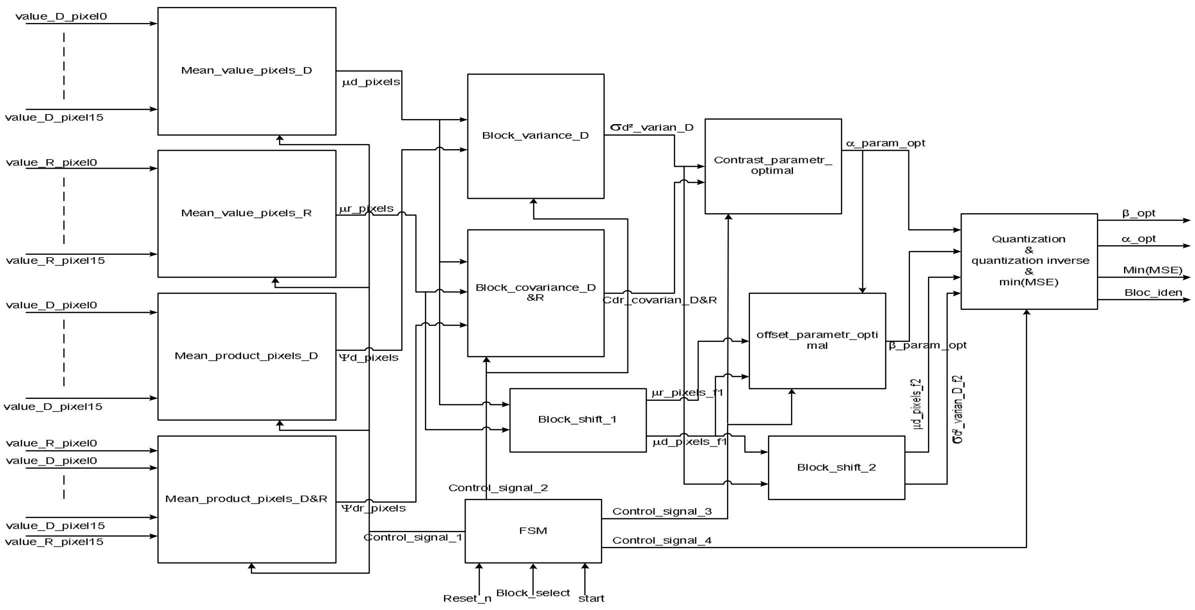

Figure 3: design of self-similarity search

- Total ALUTs: 7124
- Total register: 4559
- Total pins: 177
- Total memory bits: 571136

The proposed architecture has a frequency of 410MHz with a data flow is 13.2 Gbit/s. therefore this architecture can be used in real time video compression, because 25 frames/s needed 13 Gbit/s but we used the reduction of data by 40% proposed by [3].

5. Conclusion:

The main objective of the presented work was real-time implementation of the fractal image compression on SOC. That's why we divided the encoding algorithm into two parts: software and hardware (self-similarity search).

We conceived the self-similarity search architecture, and then we made functional and temporal simulation. Afterwards we implemented the algorithm of coding on the STRATIX chard. In experiments we have a frequency of 410MHz with a data flow of 13.2 Gbit/s.

We obtained the real time coding of video image with exploited only the ¼ totality of the resources of the STRATIX chard.

There are several open issues for future research. One is how to integrate the algorithm for estimate movement in the encoding algorithm of fractal compression, and how to improve the algorithm of fractal compression to adapt in HDTV.

.

6. References

[1] A. Martínez Ramírez, A. Díaz Sánchez, M. Linares Aranda, and J. Vega Pineda: "An Architecture for Fractal Image Compression Using Quad_tree Multiresolution", ISACS, IEEE 2004.

[2] M. Barnsly: "Fractal modelling of real word image", Spring-Velag, New York, 1998.

[3] B. Rejeb, W. Anheir : "A New approach for the Speed up of Fractal Image Coding", IEEE DSP conference, July 1997, Santorini, Greece.

[4] Saraju P. Mohanty1, Renuka Kumara C., and Sridhara Nayak : "FPGA Based Implementation of an Invisible-Robust Image Watermarking Encoder", Springer-Verlag Berlin Heidelberg 2004

[5] B. Rejeb, H. Henkelmann and W. Anheier: "Real-Time Implementation of Fractal Image Encoder using Residue Number System", IEEE NORSIG'200, Jun 2000, Sweden.

[6] Mohsen Ghazel, George H. Freeman, and Edward R. Vrscay : "Fractal-Wavelet Image Denoising Revisited IEEE Transactions on image processing, VOL. 15, NO. 9, SEPTEMBER 2006.

[7] S.Mozaffari, K.Faez, M.Ziaratban: "Character Representation and Recognition Using quadtree-based Fractal Encoding Scheme", Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05).

[8] M. Ghazel, R.K. Ward, R. Abugharbeih, E. R. Vrscay, G. Freeman: "Simultaneous Fractal Image Denoising and Interpolation", IEEE 2005

[9] Geoffrey M. Davis, Member, IEEE: "A Wavelet-Based Analysis of Fractal Image Compression", IEEE Transactions on image processing, VOL. 7, NO. 2, FEBRUARY 1998

[10] Henry, Xiao : "Fractal Compression", Related Topic Report Queen's University Kingston, Ontario, Canada April 2004